

技巧：Java垃圾收集器使用小诀窍 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/145/2021\\_2022\\_\\_E6\\_8A\\_80\\_E5\\_B7\\_A7\\_EF\\_BC\\_9AJ\\_c104\\_145540.htm](https://www.100test.com/kao_ti2020/145/2021_2022__E6_8A_80_E5_B7_A7_EF_BC_9AJ_c104_145540.htm) 垃圾收集器(Garbage Collector, GC)是现代软件虚拟机技术的重要组成部分，其设计方案对运行于虚拟机上的应用程序性能影响极大。Java虚拟机(JVM)与.net framework都提供了这一功能。下面我们简单介绍一下Java虚拟机中的垃圾收集器原理。Java的内存管理实际上就是对象的管理，其中包括对象的分配和释放。对于程序员来说，分配对象使用new关键字；释放对象时，只要将对象所有引用赋值为null。对于GC来说，当程序员创建对象时，GC就开始监控这个对象的地址、大小以及使用情况。通常，GC采用有向图的方式记录和管理堆(heap)中的所有对象。通过这种方式确定哪些对象是“可达的”，哪些对象是“不可达的”。当GC确定一些对象为“不可达”时，GC就有责任回收这些内存空间。GC在JVM中通常是由一个或一组进程来实现的，它本身也和用户程序一样占用heap空间，运行时也占用CPU。当GC进程运行时，应用程序停止运行。因此，当GC运行时间较长时，用户能够感到Java程序的停顿，另外一方面，如果GC运行时间太短，则可能对象回收率太低，这意味着还有很多应该回收的对象没有被回收，仍然占用大量内存。因此，在设计GC的时候，就必须在停顿时间和回收率之间进行权衡。根据GC的工作原理，我们可以通过一些技巧和方式，让GC运行更加有效率，更加符合应用程序的要求。以下就是一些程序设计的几点建议。1. 最基本的建议就是尽早释放无用对象的引用。大多数程序员在使用临时变

量的时候，都是让引用变量在退出活动域(scope)后，自动设置为null。

2. 尽量少用finalize函数。finalize函数是Java提供给程序员一个释放对象或资源的机会。但是，它会加大GC的工作量，因此尽量少采用finalize方式回收资源。
3. 如果需要使用经常使用的图片，可以使用soft应用类型。它可以尽可能将图片保存在内存中，供程序调用，而不引起OutOfMemory。
4. 注意集合数据类型，包括数组，树，图，链表等数据结构，对于这类对象，GC回收它们一般效率较低。如果程序允许，尽早将不用的引用对象赋为null。这样可以加速GC的工作。所以，如非必要，程序中采用的数据结构要尽可能地简单。
5. 当程序有一定的等待时间，程序员可以手动执行System.gc()，通知GC运行，但是Java语言规范并不保证GC一定会执行。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)